Loss Function

Deep Learning

Magesh anand D

The function we want to minimize or maximize is called the objective function or criterion. When we are minimizing it we may also called has the cost function, loss function, error function. We calculate a metric based on the error we observed in the network predication. The ideal state is equivalent to finding the parameters (weights and biases) that will minimize the **loss** incurred form the errors. In this way loss function help to reframe training neural network as an optimization problem.

List of loss function:

- **L1** loss function
- **L2** loss function
- Huber loss function
- Pseudo-Huber loss function
- Cross entropy loss function
- Sigmoid cross entropy loss function
- Softmax cross entropy loss function
- Hinge loss function
- Kullback-leibler divergence loss function

L1 loss function:

L1 loss function is most common function use to minimize the error in neural network for regression problem. The loss function is mean overseen the data of the absolute difference between actual values and predicted values.

$$\mathbf{L1 = N1}i\mathbf{=0}\sum N \mathbf{|y-y\hat{}}\ i\mathbf{|}$$

L1 function is not sensitive towards outliers. L1 loss function is also know has called **Least Absolute Deviation (LAD)**

L2 loss function:

L2 loss function used to minimize the error which is sum of all squared difference between actual values and predicted values. L2 function is quite sensitive to outliers.

$$L2=N1i=0\sum N(y-y\hat{}i)\ 2$$

When there is outliers the points will become the main component of the loss. When there is wrong prediction with high difference it's gonna make worst model.L2 loss function is also known as called **Least square error (LS)**

Huber loss function:

Huber loss is less sensitive to outliers in data than L2. It's basically absolute error which become quadratic when error is small. That small error has to make it quadratic on a hyper parameter δ (delta), which can be tuned. Delta $\boldsymbol{\delta}$ is consider threshold for loss function.

$$loss = \begin{cases} \frac{1}{2} * (x-y)^2 & if\ (|x-y| \leq \delta) \\ \delta * |x-y| - \frac{1}{2} * \delta^2 & otherwise \end{cases}$$

If the $(y-y\hat{}i)2$ is more than delta values the loss function is considers has the outliers. And if $(y-y\hat{})$ is lesser than delta value $½(y-y\hat{}\ )2$ going to apply has a loss function. If the $(y-y\hat{})$ is greater than delta value δ |y-y^|-½ δ^2 going to consider has a loss function. Which mean Huber loss is combination of both **L1** and **L2** depends on delta values loss function will executive.

Pseudo-Huber loss function:

The pseudo-Huber loss function can be used as a smooth approximation of the Huber loss function. It combines the best properties **L1** (Least Absolute Deviation) and **L2** (Least square error) by strongly convex when close to the target / minimum and less steep for extreme values.

$$L_\delta(a) = \delta^2 \left(\sqrt{1+(a/\delta)^2} - 1\right)$$

This steepness is controlled by the δ (delta). The pseudo-Huber loss function ensure that derivatives are continuous for all degrees.

Cross entropy loss function:

Cross-entropy or loss log, measure the performance of the classification model whose output is a probability value between 0 to 1. More specifically consider logistic regression which among to observation into two point Crosse-entropy loss increase as the predicted probability diverges from actual label.

$$L = -\sum_{i} y_i log\tilde{y}_i + (1 - y_i)\log(1 - \tilde{y}_i)$$

Always the result of cross entropy loss function value range between **(0, 1)** or **(-1, 1)** is going to consider has **yi** and **y^** is going to consider has probability values. If **yi** is 0 then first part of formula **-∑ yi (logy^)** going to consider has **0** and second part of the equation **(1-yi) log (1-y^)** will executive the loss function.

Sigmoid cross entropy loss function:

Sigmoid loss function almost similar to cross entropy function instead of log formula going to apply sigmoid function. That is major difference in this function. Others activity are more like same to cross entropy function.

$$L = -\sum_{i} y_i log\tilde{y}_i + (1 - y_i)\log(1 - \tilde{y}_i)$$

Always the result of sigmoid cross entropy loss function value range between **(0, 1)** or **(-1, 1)** is going to consider has **yi** and **y^** is going to consider has probability values. If yi is 0 then first part of formula **-∑ yi (logy^)** going to consider has 0 and second part of the equation **(1-yi) log (1-y^)** will executive the loss function. Instead of **log** the sigmoid function will execute **f(x) = 1/1+e^-x.**

Softmax cross entropy loss function:

Softmax function takes an N-dimensional vectors of real number and transform it into a vectors of real numbers in range **(0, 1)**. This function will work under the probability based for multi class classification function.

H(y,p)=−∑iyilog(pi)

Cross entropy indicates the distance between what the model believes the output distribution should be, and what the original distribution really is. Cross entropy measure is a widely used alternative of squared error. It is used when node activations can be understood as representing the probability that each hypothesis might be true, i.e. when the output is a probability distribution. Thus it is used as a loss function in neural networks which have softmax activations in the output layer.

Hinge loss function:

Hinge loss is a loss function used for binary classification problem. The hinge is used for maximum – margin classification problem mostly notably for support vector machine.

$$\ell(y) = \max(0, 1 - t \cdot y)$$

This loss function is not for a regression kind of problem. The **t** is going to consider as a number of class in classification problem. And **y** is not but the value going to receive from model. If the **t*y** is closer to 1 is minimum loss. If **t*y** not closer to 1 the maximum loss. This is known as called hinge loss function.

Kullback-leibler divergence:

The Kullback-leibler divergence it is a measure of how one probability distribution is different from second distribution. It used to compare two probability distribution and also called relative entropy. It's not distance between from two distribution often misunderstood (Divergence is not distance).Jansen-Shannon divergence calculate the distance of one probability distribution from another probability distribution. But KL divergence is not that kind of divergence.

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right)$$

|| is divergence and P and Q is are discrete probability distribution. In this loss function we have quit much freedom in our hand. With help is one-hot encoder convert the target class label to a suitable distribution that is so likely to appear out of forward pass by the model convergence. In that specific case KL divergence loss boils down to the cross entropy loss.